УДК 004.056

М.В. Гладкий, ассист.
(БГТУ, г. Минск)

РАЗРАБОТКА КОНЦЕПЦИИ СОЗДАНИЯ БЕЗОПАСНЫХ SaaS-СИСТЕМ

Контроль и управление облачными системами являются проблемой безопасности. Гарантий, что все ресурсы облака посчитаны и в нем нет неконтролируемых виртуальных машин, не запущено сторонних процессов и не нарушена конфигурация элементов облака, нет. Наиболее уязвимыми в облачной инфраструктуре являются SaaS-системы. В этом случае приложение запускается на платформе облачных вычислений и доступно через веб-браузер. Пользователь услуг не имеет возможности управлять сетью, серверами, операционными системами, хранилищем данных и некоторыми возможностями предоставляемой системы. Выбор средств и механизмов защиты полностью лежит на облачном провайдере.

Первоочередной проблемой, которая должна решаться при проектировании систем такого типа — безопасное хранение данных пользователей. Другими словами, несколько заказчиков, которые записывают и считывают информацию из одной базы данных, должны ничего не знать друг о друге.

Существует два метода для решения описанной проблемы. Первое решение, состоит в том, чтобы разделить данные разных заказчиков и хранить их в разных базах данных — мультиарендность. Мультиарендность сопоставляется с архитектурой из множественных экземпляров, в которой приложения работают одновременно с несколькими конфигурациями и наборами данных нескольких организаций, а каждая организация-клиент работает со своим экземпляром виртуального приложения, видя только свою конфигурацию и свой набор данных [1].

Второй вариант — хранить данные всех клиентов в одной базе данных в общих таблицах. Для реализации такого варианта необходимым условием будет введение дополнительного поля TenantID для разделения информации между заказчиками. К преимуществу этого подхода стоит отнести небольшую стоимость на разработку схемы базы данных и ее поддержку, а также простой способ добавления новых клиентов [2]. К недостаткам схеме можно отнести отсутствие гибкости, проблемы с резервным копированием и восстановлением данных.

При выборе решения для планирования архитектуры SaaS-систем необходимо отдавать предпочтение первому варианту: он обладает большей надежностью и гибкостью за счет простой расширяемости, масштабируемости и индивидуальности [3]. Несмотря на

высокую цену, такое решение является приемлемым для клиентов, главной целью которых является безопасность (например, банки).

С другой стороны, реализация мультиарендной архитектуры не позволяет обезопасить пользователя от таких типов угроз, как подмена идетификаторов, утечка информации или несанкционированное получение прав.

Для комплексного решения описанных проблем разработана новая концепция создания безопасных SaaS-систем, основанная на применении мультиранедной архитектуры с поддержкой версионности баз данных, механизмом безопасной передачи информации между приложением-арендодателем и приложений-арендатором, а также управление доступом на основе токенов, ролей и привилегий (рисунок 1).

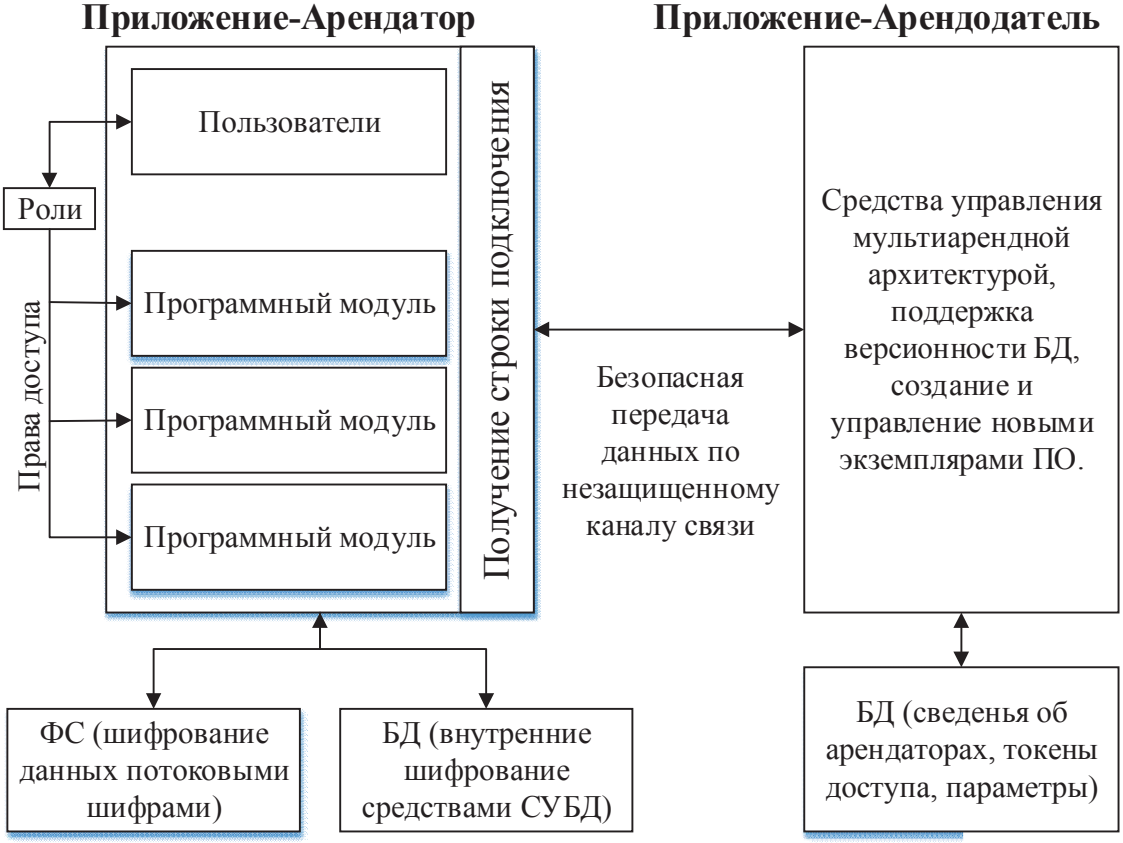

Рисунок 1 – схема концепции создания безопасных SaaS-приложений

В представленной концепции мультиарендность позволяет для каждого клиента создавать отдельные виртуальные копии программного обеспечения, что дает возможность изолированно обслуживать пользователей из разных организаций (независимых подписчиков SaaS) в рамках одного сервиса (одной инсталляции или развертывания) [1].

Данные всех клиентов сохраняются в реляционной БД с поддержкой технологий внутреннего шифрования. Функция прозрачного шифрования выполняет в реальном времени шифрование и дешифрование файлов данных и журналов в операциях ввода-вывода. При шифровании используется специальный симметричный ключ, защищенный сертификатом, который хранится в загрузочной записи базы. Примером СУБД, которая реализует описанный механизм защиты информации является MS SQL Server [4].

Услуги облачных вычислений предполагают самообслуживание, что может создать путаницу в управлении обновлениями [2]. Для предотвращения этого процесса используются механизмы для работы с базами данных различных версий в зависимости от версии используемого виртуального приложения. В случаи выпуска пакета обновлений, меняющего структурную схему базы данных (добавление, имение, удаление таблиц, колонок или иные операции), администратор сам решает, стоит ли делать обновление текущего состояния базы данных до новой версии.

При передаче информации между составными частями SaaS-систем всегда есть вероятность появлении ситуации, когда атакующий способен читать и видоизменять сообщения, которыми обмениваются участники системы, причем ни один из последних не может догадаться о его присутствии в канале. Для защиты от данного типа атаки разработан алгоритм обмена информацией, основанный на криптографическом протоколе Диффи-Хеллмана и алгоритме шифрования RSA (рисунок 2).

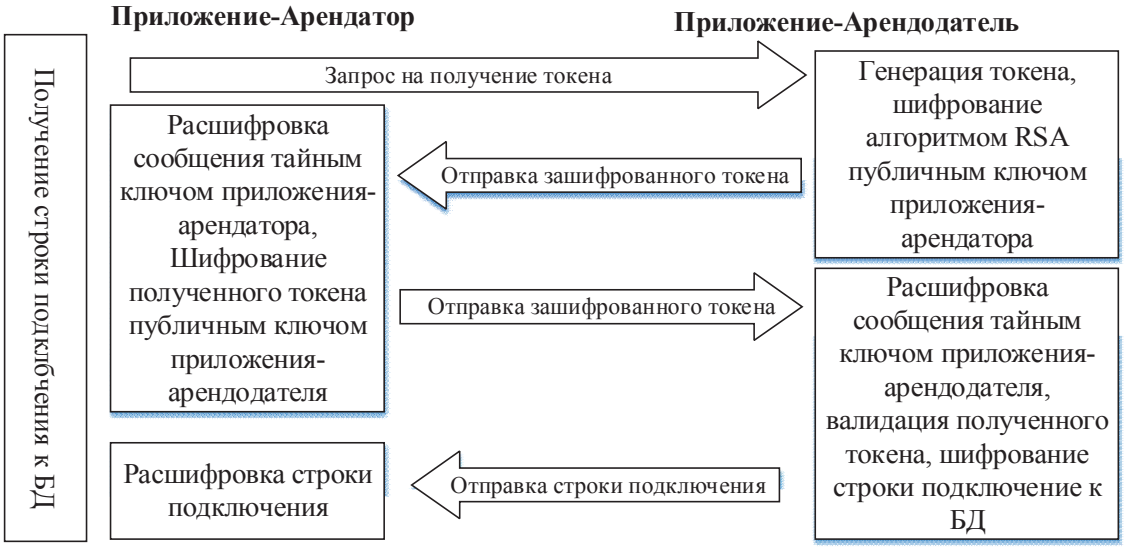

Рисунок 2 – Схема передачи информации в SaaS-системах

В случае успешной верификации токена доступа пользователю передается строка подключения к базе данных в зашифрованном виде. Для получения доступа к модулям программного обеспечения необходимо использовать механизм разграничения прав на основе привилегий и ролей. Он имеет свои особенности в виде динамической связи ролей и наборов привилегий и возможности привязки пользователя сразу к нескольким ролям. Такой подход позволяет гибко настраивать права доступа для выбранного пользователя, создавать множество ролей с разными наборами привилегий.

Если в рамках функционирования системы необходимо сохранять какую-либо информацию не в базе данных, а в облачном хранилище, предоставляемым провайдером, то эти данные шифруются с помощью поточного шифра (например, RC4 или SEAL). Таким образом, в случае хищения файлов, злоумышленники не смогут их прочитать.

Разработанная концепция создания безопасных SaaS-приложений универсальна и может быть использована при реализации облачных систем различной тематической направленности. Ее реализация возможна на многих языках веб-программирования, таких как c#, Java EE, Python или PHP. Особенностями разработанной концепции являются использование мультиарендной архитектуры, позволяющая виртуальным копиям приложений работать независимо друг от друга, механизмов управления экземплярами приложений, версиями баз данных, алгоритмом безопасной передачи информации по незащищенным каналам связи, системой разграничения прав доступа, основанной на динамической связи ролей и привилегий.

ЛИТЕРАТУРА

1. Гладкий, М.В. Безопасность приложений на платформах облачных вычислений / М.В. Гладкий // Информационные технологии: тезисы 79-й науч.-техн. конференции профессорско-преподавательского состава, научных сотрудников и аспирантов, Минск, 2-6 февраля 2015 г. – Минск: БГТУ, 2015. – 73с.

2. Шаньгин В. Ф. Защита информации в сети – анализ технологий и синтез решений / В.Ф. Шаньгин, С.Д. Рябко, А.В. Галицкий. – Москва: ДМК Пресс, 2004. – С. 614–616.

3. Риз, Д. Облачные вычисления. / Д. Риз. – Санкт-Петербург: БХВ-Петербург, 2011. – С. 280–288.

Александров, А. Как предотвратить вторжение: второй уровень защиты. / А. Александров. – Москва: BYTE, 2003. – 36с.